24

Motion-Tracking and Speech Recognition for Hands-Free Mouse-Pointer Manipulation

Frank Loewenich and Frederic Maire
Queensland University of Technology
Australia

1. Introduction

The design of traditional interfaces relies on the use of mouse and keyboard. For people with certain disabilities, however, using these devices presents a real problem.
In this chapter we describe a graphical user interface navigation utility, similar in functionality to the traditional mouse pointing device. Movement of the pointer is achieved by tracking the motion of the head, while button-actions can be initiated by issuing a voice command. Foremost in our mind was the goal to make our system easy to use and affordable, and provide users with disabilities with a tool that promotes their independence and social interaction.
The chapter is structured as follows. Section 2 provides an overview of related work. Section 3 describes our proposed system architecture, the face detection and feature tracking algorithms, as well as the speech recognition component. Section 4 contains experimental results and Section 5 discusses future work. We conclude the chapter in Section 6.

2. Research motivation and related work

Persons with disabilities are often unable to use computers. This is because they are either unable to find a suitable means of interaction or they simply cannot afford commercial solutions. We also found that available solutions do not promote the individual's sense of independence, as they require a third party to attach markers at various points of their anatomy. Our work addresses these shortcomings by providing a non-intrusive, reliable, inexpensive and robust visual tracking system. It allows persons, who may have disabilities ranging from not being able to use their hands to severe cases where the person is only able to move their head, to navigate and manipulate the graphical user interface using head movements and speech.
Research into assistive technologies is ongoing and in (Evans et al., 2000), the authors describe a head-mounted infrared-emitting control system that is a 'relative' pointing device and acts like a joystick rather than a mouse. In (Chen et al., 1999) a system containing an infrared transmitter was described. The transmitter was mounted on to the user's eyeglasses, along with a set of infrared receiving modules that substitute the keys of a computer keyboard, and a tongue-touch panel to activate the infrared beam. In (Atyabi et al., 2006) the authors describe a system for translating a user's motion to mouse movements.

Source: Speech Recognition, Technologies and Applications, Book edited by: France Mihelič and Janez Žibert,
ISBN 978-953-7619-29-9, pp. 550, November 2008, I-Tech, Vienna, Austria

Their tracking algorithm relies on detecting specific features such as the eyes or nose to follow head movements across multiple frames.
There are also various commercial mouse alternatives available today. NaturalPoint (NaturalPoint, 2006) markets several head-tracking-based mouse alternatives on their web site. While the benefits are real, these devices still require the user to attach markers either to the head or glasses. Other systems use infrared emitters that are attached to the user's glasses, head-band, or cap. Some systems, for example the Quick Glance system by EyeTech Digital Systems (EyeTech, 2006), place the transmitter over the monitor and use an infrared-reflector that is attached to the user's forehead or glasses. Mouse clicks are generated with a physical switch or a software interface.

3. System architecture

We have implemented a prototype of our system that requires only a web camera and microphone, which fully emulates the functionality normally associated with a mouse device. In this section we provide a short overview of the image-processing algorithms and speech-interaction technologies the system is based on.
The system consists of two signal-processing units and interpretation logic (see Figure 1). Image processing algorithms are applied to the video stream to detect the user's face and follow tracking points to determine head movements. The audio stream is analyzed by the speech recognition engine to determine relevant voice commands. The interpretation logic in turn receives relevant parameters from the signal-processing units and translates these into on-screen actions by the mouse pointer.

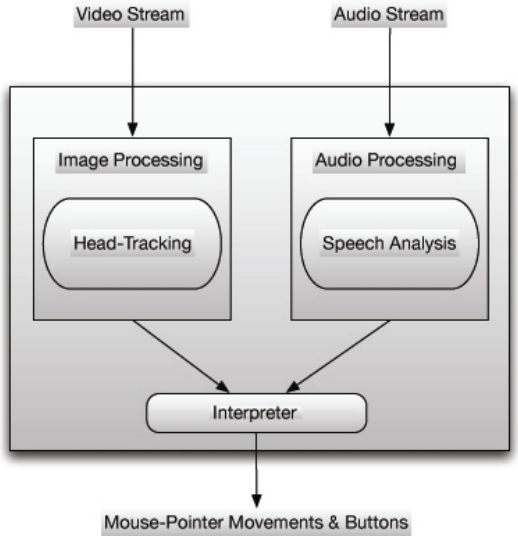

Fig. 1. High-level system overview.

3.1 Face detection

The face-detection component is fundamental to the functioning of our head-tracking system and is based on the Haar-Classifier cascade algorithm. This algorithm was first

introduced by Viola and Jones (Viola & Jones, 2001). It is appearance-based and uses a boosted cascade of simple feature classifiers, providing a robust framework for rapid visual detection of frontal faces in grey scale images. The process of face detection is complicated by a number of factors. Variations in pose (frontal, non-frontal and tilt) result in the partial or full occlusion of facial features. Beards, moustaches and glasses also hide features. Another problem is partial occlusion by other objects. For example, in a group of people some faces may be partially covered by other faces. Finally, the quality of the captured image needs consideration.

The algorithm is based on AdaBoost classification, where a classifier is constructed as a linear combination of many simpler, easily constructible weak classifiers. For AdaBoost learning algorithm to work each weak classifier is only required to perform slightly better than a random guess. For face detection a set of Haar wavelet-like features is utilized. Classification is based on four basic types of scalar features, proposed by Viola and Jones for the purpose of face detection. Each of these features has a scalar value that can be computed efficiently from the integral image, or summed area table. This set of features has recently been extended to deal with head rotation (Lienhart & Maydt, 2002). Weak classifiers are cascaded into a collection of weak classifiers (weak learners) to form a stronger classifier. AdaBoost is an adaptive algorithm to boost a sequence of classifiers, in that the weights are updated dynamically according to the errors in previous learning cycles. The algorithm employed by Viola & Jones (Viola & Jones, 2001) has a face detection cascade of 38 stages with 6000 features. According to Viola & Jones, the algorithm nevertheless achieved fast average detection times. On a difficult dataset, which contained 507 faces and 75 million sub-windows, faces are detected using an average of 10 feature evaluations per sub-window. As a comparison, Viola & Jones show in experiments that their system is 15 times faster than a detection system implemented by Rowley et al. (Rowley et al., 1998). A strong classifier, consisting of a cascade of weak classifiers is shown in Figure 3, where blue icons are non-face and red icons are face images. The aim is to filter out the non-face images, leaving only face images. The cascade provides higher accuracy over a single, weak classifier.

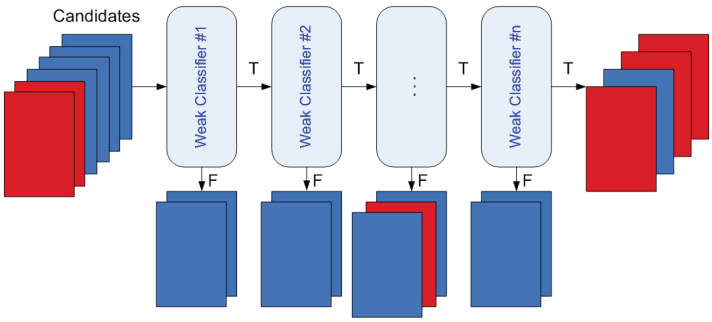

Fig. 3. Weak classifiers are arranged in a cascade. Note that some incorrect classifications may occur, as indicated in the diagram.

3.2 Head-Tracking

Our optical tracking component uses an implementation of the Lucas-Kanade optical flow algorithm (Lucas & Kanade, 1981), which first identifies and then tracks features in an image.

These features are pixels whose spatial gradient matrices have a large enough minimum eigenvalue. When applied to image registration, the Lucas-Kanade method is usually carried out in a coarse-to-fine iterative manner, in such a way that the spatial derivatives are first computed at the coarse scale in the pyramid, one of the images is warped by the computed deformation, and iterative updates are then computed at successively finer scales.

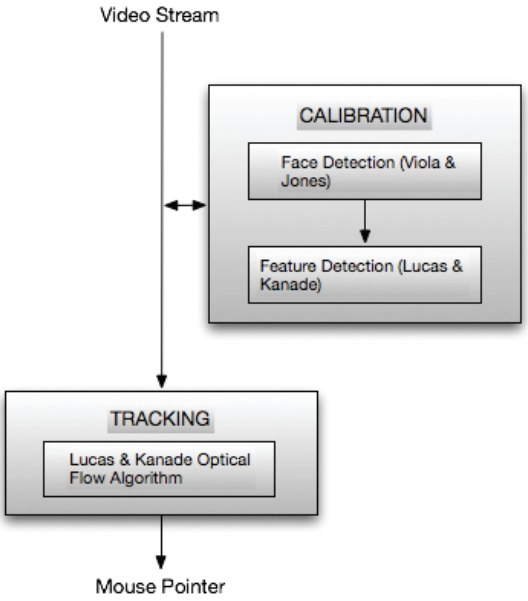

Fig. 4. The head-tracking component. After initial calibration, the video stream is processed in real-time by the tracking algorithm.

Our algorithm restricts the detection of feature points to the face closest to the computer screen to exclude other people in the background. Before tracking is initiated, feature points are marked with a green dot (Figure 5), and are still subject to change.

Fig. 5. The face is detected using the Haar-Classifier cascade algorithm and marked with a yellow square. Green dots mark significant features identified by the Lucas-Kanade algorithm. The image on the right demonstrates performance in poor lighting conditions.

Once tracking has been started, feature points are locked into place and marked with a red dot. It should be noted that the marking of the features and the face is for debugging and demonstration purposes only. Tracking is achieved by calculating the optical flow between two consecutive frames to track the user's head movement, which is translated to on-screen movements of the mouse pointer (see Figure 6).

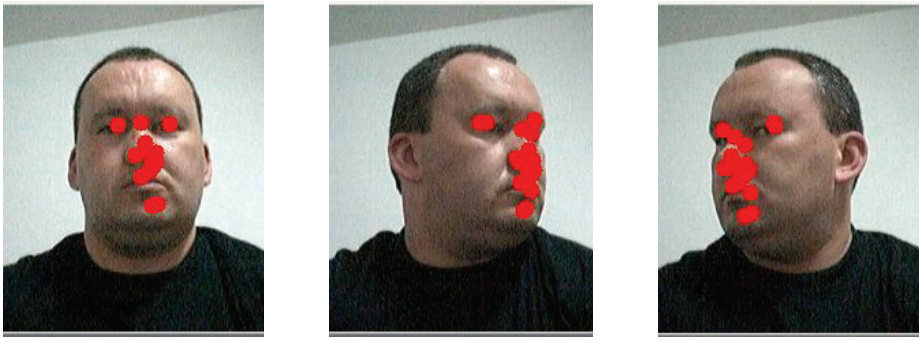

Fig. 6. Feature extraction using modified Lucas-Kanade algorithm.

The user can choose between two operational modes for translating head movements to on-screen mouse pointer movements. Selection of the desired mode may be accomplished in real-time by issuing a voice command. These modes are:

Relative Mode

This mode simulates the joystick control of a mouse pointer. If the system detects deviation of the tracking points from their original position above a certain threshold, the mouse pointer is moved in the given direction by a single pixel. Movement in this direction continues as long as the deviation of the tracking points is maintained. However, the rate of pixels being moved is steadily increased relative to the amount of time elapsed since the movement was initiated. Should the movement be interrupted, the rate of movement is reset to a single pixel. This mode is especially useful where fine control of the mouse pointer is required, and where navigation of the whole screen is still necessary. This makes the system ideal for disabled users who have to make precise on-screen movement, such as artists or engineers.

Absolute Mode

This closely resembles mouse pointer control associated with mouse hardware devices. In this mode, the distance of the tracking points from their original location is translated to the location of the mouse pointer from the center of the screen. The distance the mouse pointer will move away from the center of the screen depends on the resolution setting reported by the operating system.

3.3 Speech recognition

Speech recognition technology has progressed to a level, where users can to some extend control actions on the computer using voice commands. The most popular speech recognition technique is based on hidden Markov models (HMM). HMM is a doubly stochastic model, where the underlying phoneme string and frame-by-frame, surface acoustic realizations are represented as probabilistic Markov processes. HMM systems

classify speech segments using dynamic programming. As an alternative to the frame-by-frame approach, similar speech recognition performance has been achieved by first identifying speech segments, classifying the segments and then using segment scores to recognise words. Speech recognition has matured and with the ability to train the parameters of the model using training data to gain optimal performance, it represents a powerful solution (Rabiner & Juang, 1986).

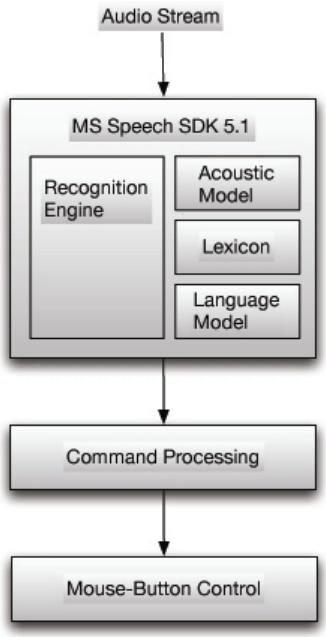

Fig. 7. The speech-recognition components of our head-tracking system.

Our prototype system utilizes the Microsoft Speech SDK 5.1, which is available as a free download from Microsoft (see Figure 7). By issuing the relevant voice command, the user may at any time perform common tasks, such as click-drag and drag-drop operations. The need for double-clicking, for example, represents a great challenge for people with reduced dexterity and motor control in their hands. Our system not only enables the user to execute single- or double-clicks, but also more complex operations. For example, a click-drag operation can be initiated with a vocal command or a file may be picked up and dropped on another folder by issuing a simple sequence of two commands (pick-up and release) in combination with head movements (carrying the object to another location).
Voice commands used to activate buttons are:

- "Click" – single left click
- "Double – double left click
- "Right" – single right click
- "Hold" – single left click and hold (click-drag)
- "Drop" – release the left mouse button after performing a click-drag operation

The commands may be changed to suit the individual user. There is also scope to extend the set of commands for added functionality.

4. Experimental results

In preliminary trials, the effectiveness of our system was tested with a group of 10 volunteers. Each of the subjects had previously used a computer, and was thus able to comment on how our system compares to using a traditional mouse pointing device. Each user was given a brief tutorial on how to use the system, and then allowed to practice moving the cursor for one minute. After the practice period, each user was asked to play a video game (Solitaire). The one minute training time was perceived as sufficient to introduce the features of the system.

Subsequent interviews revealed that users preferred to learn on-the-job, while using our mouse-replacement system to play a computer game. Task completion times were similar to using a traditional mouse device. However, users commented positively on the fact that they were able to control the mouse pointer using head movements, while their hands remained free to perform other tasks. Also positive was the short amount of time users required to become acquainted with the system. They found that using head movements for mouse control quickly become second-nature and issuing voice commands requires no effort at all. All users commented on the possibility of extending the set of voice commands to add functionality to the system.

5. Future work

A future implementation of our system could further extend the speech recognition component, allowing the user, for example, to open applications using vocal commands would eliminate much navigating through menu structures. The speech synthesis component could be extended to provide contextual feedback in the form of reading to the user the names of windows or icons the mouse pointer is currently resting on.

Although the system has been developed and tested on the Microsoft Windows operating system, it is possible to port this technology to Apple and Linux/Unix operating systems. In particular deployment on an open-source platform would open the technology up to a much wider user base. Developers can now develop for Windows, and expect their .NET application to run on Apple and Linux operating systems (Easton & King, 2004). Recent work by the 'Mono Project' and 'Portable .NET'-project has contributed to making .NET code truly portable, presenting the possibility to use the system on these platforms with minimal change.

6. Conclusion

The prototype system has proven to be robust, being able to tolerate strong variations in lighting conditions. We see potential for our approach being integrated in interfaces designed specifically for users with disabilities. Especially attractive should be the fact that it provides a low-cost means of human-computer interaction requiring the most basic of computer hardware and its reliance upon established approaches in the areas of computer vision and speech interaction. It may also prove useful for other applications, for example, where it is necessary to activate controls on a computer interface while at the same time performing precision work using both hands. Another application area could be computer games. Furthermore, the modular architecture of the system allows for ready integration in any number of software projects requiring a low-cost and reliable head-tracking component.

7. References

Atyabi, M., Hosseini, M. S. K., & Mokhtari, M. (2006). The Webcam Mouse: Visual 3D Tracking of Body Features to Provide Computer Access for People with Severe Disabilities. *Proceedings of the Annual India Conference*, pp. 1-6, ISBN: 1-4244-0369-3 , New Delhi, September 2006

Camus, T. A. (1995). Real-time optical flow. *Technical Report CS-94-36*. The University of Rochester. Rochester, New York

Chen Y. L., Tang F. T., Chang W. H., Wong M. K., Shih Y. Y., & Kuo T. S. (1999). The new design of an infrared-controlled human-computer interface for the disabled, *IEEE Trans. Rehab. Eng.*, Vol. 7, No. 4, December 1999, 474–481, ISSN: 1063-6528

Cole, R., Mariani, J., Uszkoreit, H., Varile, G., B., Zaenen, A., & Zampolli, A. (1998). *Survey of the State of the Art in Human Language Technology,* Cambridge University Press, ISBN-10: 0521592771, Cambridge

Easton, M. J. & King, J. (2004). *Cross-Platform .NET Development: Using Mono, Portable .NET, and Microsoft .NET*. Apress Publishers, ISBN: 1590593308, Berkeley

Evans, D. G., Drew, R., & Blenkhorn, P. (2000). Controlling mouse pointer position using an infrared head-operated joystick. *Rehabilitation Engineering, IEEE Transactions on [see also IEEE Trans. on Neural Systems and Rehabilitation],* Vol. 8, No. 1, March 2000, 107-117, ISSN: 1063-6528

EyeTech Digital Systems Inc. (2006). On-line Product Catalog: Retrieved May 29 from http://www.eitetechds.com/products.htm

Lienhart, R. & Maydt, J. (2002). An extended set of Haar-like features for rapid object detection. *Proceedings of the International Conference on Image Processing (ICIP)*, pp. I-900- I-903, September 2002, IEEE, Rochester, New York, USA

Loewenich, F., & Maire, F. (2006). A Head-Tracker Based on the Lucas-Kanade Optical Flow Algorithm. In: *Advances in Intelligent IT - Active Media Technology 2006,* Li, Y., Looi, M. & Zhong, N. (Ed.), Vol. 138, pp. 25-30, IOS Press, ISBN: 1-58603-615-7, Amsterdam, Netherlands

Lucas, B. D., & Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pp. 674—679, Vancouver

NaturalPoint Inc. (2006). On-line Product Catalog: Retrieved May 29 from http://www.naturalpoint.com/trackir/02-products/product-TrackIR-4-PRO.html

Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. ASSP Magazine, IEEE, Vol. 3, No. 1, 4-16, ISSN: 0740-7467

Viola, P., & Jones, J. J. (2001). Robust Real-Time Face Detection. *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pp. 122-130, ISBN:0-7803-7965-9, July 2003, IEEE Computer Society, Washington, DC, USA

Viola, P., & Jones, J. J. (2004) Robust Real-Time Face Detection. *International Journal of Computer Vision*, Vol. 57, No. 2, May 2004, 137-154, ISSN: 0920-5691